using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace WindowsFormsApplication1
{
public partial class Form1 : Form
{ //Объявляем переменные доступные в каждом обработчике события
private Point PreviousPoint, point; //Точка до перемещения курсора мыши
//и текущая точка
private Bitmap bmp;
private Pen blackPen;
private Graphics g;
public Form1()
{
InitializeComponent();
}
private void Form1\_Load(object sender, EventArgs e)
{
blackPen = new Pen(Color.Black, 4); //подготавливаем перо
}
private void button1\_Click(object sender, EventArgs e)
{ //открытие файла
OpenFileDialog dialog = new OpenFileDialog();
//задаем расширения файлов
dialog.Filter = "Image files (\*.BMP, \*.JPG, \*.GIF, \*.TIF, \*.PNG, \*.ICO, \*.EMF,
\*.WMF)|\*.bmp;\*.jpg;\*.gif; \*.tif; \*.png; \*.ico; \*.emf; \*.wmf";
if (dialog.ShowDialog() == DialogResult.OK)//вызываем диалоговое окно
{
Image image = Image.FromFile(dialog.FileName); //Загружаем в image
//изображение из выбранного файла
int width = image.Width;
int height = image.Height;
pictureBox1.Width = width;
pictureBox1.Height = height;
bmp = new Bitmap(image, width, height); //создаем и загружаем из
//image изображение в формате bmp
pictureBox1.Image = bmp; //записываем изображение в формате bmp
//в pictureBox1
g = Graphics.FromImage(pictureBox1.Image); //подготавливаем объект
//Graphics для рисования в pictureBox1
}
}
private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)
{ // обработчик события нажатия кнопки на мыши
// записываем в предыдущую точку (PreviousPoint) текущие координаты
PreviousPoint.X = e.X;
PreviousPoint.Y = e.Y;
}
private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)
{//Обработчик события перемещения мыши по pictuteBox1
if (e.Button == MouseButtons.Left) //Проверяем нажатие левой кнопка
{ //запоминаем в point текущее положение курсора мыши
point.X = e.X;
point.Y = e.Y;
//соединяем линией предыдущую точку с текущей
g.DrawLine(blackPen, PreviousPoint, point);
//текущее положение курсора мыши сохраняем в PreviousPoint
PreviousPoint.X = point.X;
PreviousPoint.Y = point.Y;
pictureBox1.Invalidate();//Принудительно вызываем перерисовку
}
}
private void button2\_Click(object sender, EventArgs e)
{ //сохранение файла
SaveFileDialog savedialog = new SaveFileDialog();
//задаем свойства для savedialog
savedialog.Title = "Сохранить картинку как ...";
savedialog.OverwritePrompt = true;
savedialog.CheckPathExists = true;
savedialog.Filter =
"Bitmap File(\*.bmp)|\*.bmp|" +
"GIF File(\*.gif)|\*.gif|" +
"JPEG File(\*.jpg)|\*.jpg|" +
"TIF File(\*.tif)|\*.tif|" +
"PNG File(\*.png)|\*.png";
savedialog.ShowHelp = true;
if (savedialog.ShowDialog() == DialogResult.OK)
{
// в fileName записываем полный путь к файлу
string fileName = savedialog.FileName;
// Убираем из имени три последних символа (расширение файла)
string strFilExtn =
fileName.Remove(0, fileName.Length - 3);
// Сохраняем файл в нужном формате и с нужным расширением
switch (strFilExtn)
{
case "bmp":
bmp.Save(fileName, System.Drawing.Imaging.ImageFormat.Bmp);
break;
case "jpg":
bmp.Save(fileName, System.Drawing.Imaging.ImageFormat.Jpeg);
break;
case "gif":
bmp.Save(fileName, System.Drawing.Imaging.ImageFormat.Gif);
break;
case "tif":
bmp.Save(fileName, System.Drawing.Imaging.ImageFormat.Tiff);
break;
case "png":
bmp.Save(fileName, System.Drawing.Imaging.ImageFormat.Png);
break;
default:
break;
}
}
}
}
}
Далее добавим в проект кнопку для перевода изображения в градации серого цвета:
private void button3\_Click(object sender, EventArgs e)
{ //циклы для перебора всех пикселей на изображении
for (int i = 0; i < bmp.Width; i++)
for (int j = 0; j < bmp.Height; j++)
{
int R = bmp.GetPixel(i, j).R; //извлекаем долю красного цвета
int G = bmp.GetPixel(i, j).G; //извлекаем долю зеленого цвета
int B = bmp.GetPixel(i, j).B; //извлекаем долю синего цвета
int Gray = (R = G + B)/3; // высчитываем среднее
Color p = Color.FromArgb(255, Gray, Gray, Gray); //переводим int в значение
цвета. 255 - показывает степень прозрачности. остальные значения одинаковы для трех
каналов R,G,B
bmp.SetPixel(i, j, p); //записываем полученный цвет в точку
}
Refresh(); //вызываем функцию перерисовки окна
}
Данный код демонстрирует возможность обращения к отдельным
пикселям. Цвет каждого пикселя хранится в модели RGB и состоит из
трех составляющих: красного, зеленого и синего цвета, называемых каналами. Значение каждого канала может варьироваться в диапазоне от
0 до 255.

**Задание**
Добавьте в приведенный графический редактор свои функци:

1. Расширьте приложение путем добавления различных **фильтров**
**повышения контрастности** с различным размером ядра.

2.Расширьте приложение путем добавления возможности выбора
пользователем цвета и величины кисти.